AI-Based Public Safety Monitoring & Risk Detection System

Project Report

Project Description

An intelligent crowd behavior analysis system that automatically detects anomalous patterns in public spaces to prevent safety incidents like stampedes, panic situations, and crowd surges. The system processes video feeds in real-time to identify dangerous crowd dynamics without compromising individual privacy.

Key Capabilities:

- **Crowd-level person detection** using state-of-the-art YOLOv11 deep learning model
- **Motion pattern analysis** through dense optical flow computation
- **Statistical anomaly detection** using adaptive baseline learning and z-score analysis
- **Multi-tier risk classification** (HIGH/MEDIUM/LOW) with confidence scoring
- **Real-time alert generation** with intelligent cooldown mechanisms
- **Privacy-preserving design** - no face recognition or individual tracking

Problem Solved:

Traditional CCTV monitoring relies on human operators who cannot effectively monitor multiple camera feeds simultaneously and may miss critical early warning signs. This AI system provides:

- **Automated 24/7 monitoring** with no human fatigue
- **Early warning detection** before situations escalate
- **Objective analysis** using statistical methods
- **Scalability** across multiple camera feeds
- **Fast response times** (processes at 15-60 FPS depending on hardware)

Tech Stack Used

Core Technologies:

- **Python 3.8+** - Primary programming language
- **YOLOv11 (Ultralytics)** - Latest object detection model (2024-2025)
- **OpenCV 4.x** - Computer vision and video processing
- **NumPy** - Numerical computations and array operations
- **PyTorch** - Deep learning framework (backend for YOLO)

Additional Libraries:

- **SciPy** - Statistical analysis and distance calculations
- **Collections (deque)** - Efficient temporal data storage
- **datetime** - Timestamp generation for alerts

Algorithms Implemented:

1. **YOLOv11** - Person detection (pre-trained on COCO dataset)
2. **Farneback Optical Flow** - Dense motion field estimation
3. **Z-Score Statistical Analysis** - Anomaly detection
4. **Shannon Entropy** - Motion chaos quantification
5. **Spatial Clustering** - Crowd distribution analysis
6. **Temporal Pattern Recognition** - Time-series analysis

Hardware Support:

- **CPU Mode**: Works on any modern processor (Intel/AMD)
- **GPU Mode**: CUDA-enabled NVIDIA GPUs (optional)
- **Memory**: 4-8GB RAM recommended

Setup Instructions

Step 1: Install Python

Ensure Python 3.8 or higher is installed:

bash

python --version

Step 2: Install Required Libraries

bash

 # Install all dependencies

pip install ultralytics opencv-python numpy torch scipy

 # Or use requirements.txt

pip install -r requirements.txt

requirements.txt:

ultralytics>=8.0.0

opencv-python>=4.8.0

numpy>=1.24.0

torch>=2.0.0

scipy>=1.10.0

Step 3: Prepare Video File

Place your crowd video file in the same directory

Step 4: Run the System